**Table of Contents**

[Important rules to manage and save scripts 1](#__RefHeading___Toc13799_3670047285)

[Scripts for processing data 2](#__RefHeading___Toc12855_3670047285)

[From GSM to SRR 2](#__RefHeading___Toc12857_3670047285)

[From SRR to fastq file and gzip 2](#__RefHeading___Toc12863_3670047285)

[Extract infor from NCBI: link GSM with Pubmed ID and SRR 2](#__RefHeading___Toc12869_3670047285)

[Extract title of the articles based on PubmedID 3](#__RefHeading___Toc12871_3670047285)

[Change chromosome name 3](#__RefHeading___Toc13152_3670047285)

[Working in interaction mode on Computerome 3](#__RefHeading___Toc21465_3670047285)

[Experience for using SpikeInFree 3](#__RefHeading___Toc12859_3670047285)

[Datasets the authors used in SpikeInFree 4](#__RefHeading___Toc21467_3670047285)

[Density plot (used in SpikeinFree paper): 4](#__RefHeading___Toc21469_3670047285)

[How to run an R-script on Computerome 4](#__RefHeading___Toc20887_3670047285)

[Some Linux commands that are useful 5](#__RefHeading___Toc20889_3670047285)

[Working with gtf file 5](#__RefHeading___Toc13154_3670047285)

[htseq-count 5](#__RefHeading___Toc13156_3670047285)

[Broad Institute manual 5](#__RefHeading___Toc13492_3670047285)

[Samtools sort command creates a lot of tmp files 5](#__RefHeading___Toc13158_3670047285)

[Resource to learn sed/awk quickly 5](#__RefHeading___Toc13160_3670047285)

[Change cache folder for using sra toolkits 6](#__RefHeading___Toc13162_3670047285)

[Proper way to download paired-end raw sequencing data (fastq files) from NCBI 6](#__RefHeading___Toc13164_3670047285)

[Notes that include a lot of concepts 6](#__RefHeading___Toc13494_3670047285)

[Rules to process TT-seq data 6](#__RefHeading___Toc13801_3670047285)

[Sam files 7](#__RefHeading___Toc13917_3670047285)

[Duplicated reads 8](#__RefHeading___Toc13919_3670047285)

[Understanding FastQC report from manual 8](#__RefHeading___Toc15847_3670047285)

[Notes about genomic interval 10](#__RefHeading___Toc15849_3670047285)

[Example of features (from GFF parser): 10](#__RefHeading___Toc15851_3670047285)

[Some notes about sed/awk commands 10](#__RefHeading___Toc17139_3670047285)

[RNA-seq revision 14](#__RefHeading___Toc17633_3670047285)

[RNA-seq workflow and applications 15](#__RefHeading___Toc20618_3670047285)

[Long non-coding RNA (lncRNA) 16](#__RefHeading___Toc20601_3670047285)

[Other useful sources to learn bioinformatics 17](#__RefHeading___Toc20603_3670047285)

[Flag information of reads 17](#__RefHeading___Toc20605_3670047285)

[Important note about installing with global and local 17](#__RefHeading___Toc20607_3670047285)

[Notes about using virtual environment 18](#__RefHeading___Toc20609_3670047285)

[Review about Anaconda, Conda 19](#__RefHeading___Toc20613_3670047285)

[Managing environment variables 19](#__RefHeading___Toc20615_3670047285)

## Important rules to manage and save scripts

+Script name must have created date, dataset name, description

+If script need to be modified, generate similar one with version number. Later scripts must use bigger version number

+Don’t delete script that lead to losing track

+Never overwrite script files

+Never delete log files

## Scripts for processing data

### From GSM to SRR

GSMs="$(cat GSM\_IDs.csv | cut -f1 )"

for GSM in $GSMs ;

do

echo $GSM retrieved from NCBI

SRR="$(esearch -db sra -query $GSM |efetch -format docsum |xtract -pattern DocumentSummary -element Run@acc)"

echo $SRR is fastq-dump

echo "$GSM --> $SRR" >> GSM.and.SRR.txt

echo fastq-dump done.....

done

### **From SRR to fastq file and gzip**

SRRs="$(cat SRR3.csv | cut -f1 )"

for SRR in $SRRs ; do

echo $SRR is fastq-dump

prefetch -v $SRR

fastq-dump -A $SRR -O sra\_fastq

echo fastq-dump done.....

gzip "/home/projects/cu\_10056/data/Yang/MitoNormalization/sra\_fastq/${SRR}.fastq"

done

### **Extract infor from NCBI: link GSM with Pubmed ID and SRR**

- wget to download the html file of GSM: <https://www.ncbi.nlm.nih.gov/gds/?term=GSM1464990> with --no-check-certificate

- Manually see the pattern of the file, and extract:

+ Source names

+ Organism

+ Characteristics

+ Extracted molecule

+ Treatment protocol

+ Data processing

### Extract title of the articles based on PubmedID

pubs="$(cat DataSetList.csv | cut -f1 )"

for p in $pubs

do

echo $p

out="$(esearch -db pubmed -query $p | efetch -format docsum | xtract -pattern DocumentSummary -element Title)"

temp="${p} ${out}"

echo $temp >> pubs.cs

done

### Change chromosome name

-Lệnh sed đổi chromosome name trong bam file:

sed -e '/^@SQ/s/SN\:/SN\:chr/' -e '/^[^@]/s/\t/\tchr/2'

Ý nghĩa:

-e: có thể thay thế nhiều lần

-/^@SQ/s/SN\:/SN\:chr/: tìm từ đầu dòng, chỗ nào có @SQ, theo sau là space, tiếp theo là SN: thì sẽ thay bằng SN:chr. Như vậy nếu ta có SN:chr10 thì nó sẽ thành SN:chrchr10 → không đúng ý định. Tương tự, SN:MT sẽ thành SN:chrMT.

-'/^[^@]/s/\t/\tchr/2': tìm dòng ko bắt đầu với @, sau đó thay thế tab thứ 2 bằng tab+chr (cái này là do chromosome name nằm ở tab 2 này, ta muốn 2 thành chr2) → ko đúng ý ta muốn là đổi riêng cho MT thành chrM

## Working in interaction mode on Computerome

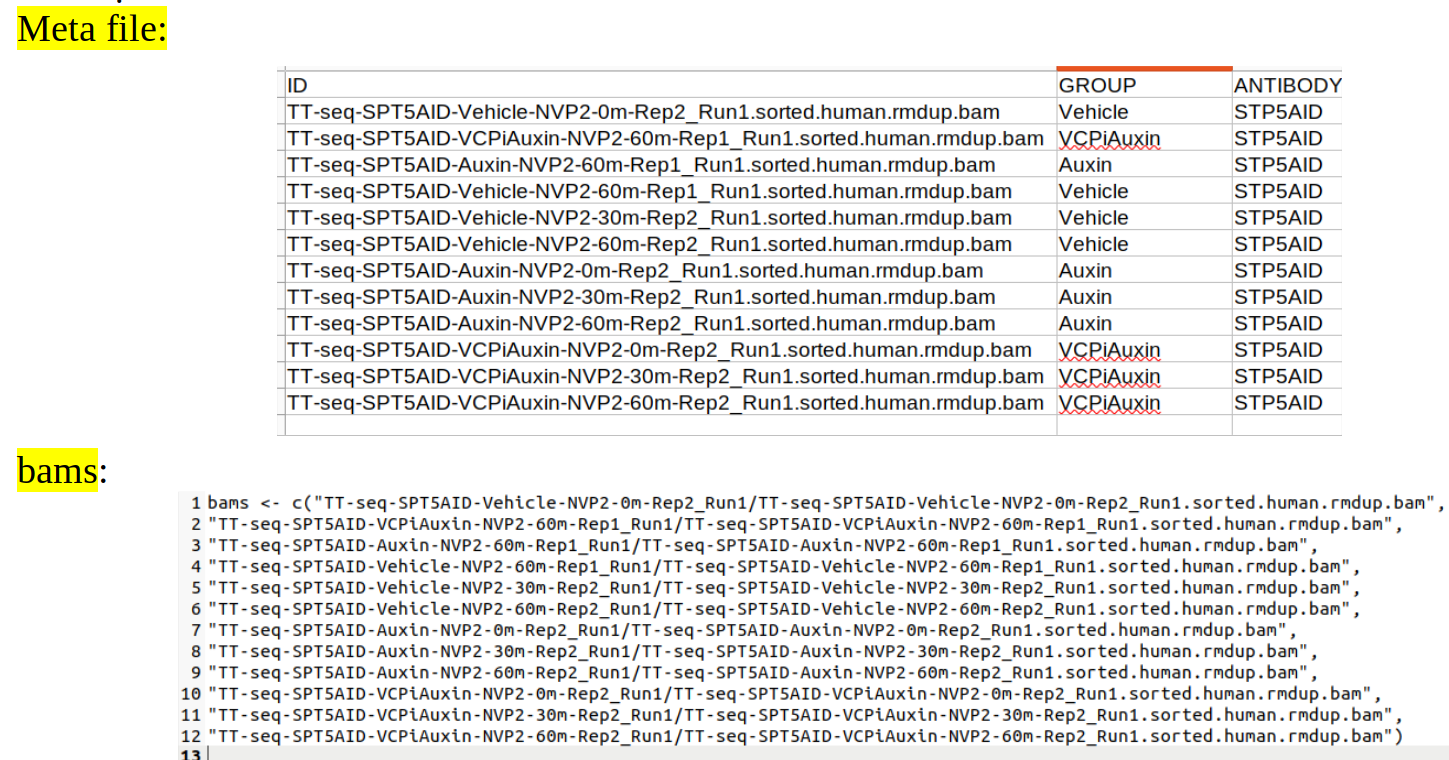
-Ko xem được job đang chạy/cửa sổ đang chạy là interaction tương ứng với job\_id nào

-Rất dễ nhầm interaction model với login mode

## Experience for using SpikeInFree

-It should be noted that meta.txt file must use tab delimiters

-In meta.txt file, samples should be grouped based on antibody, the first column is bam filename and will be used as ID (not include the path). Example in the image below:



-It is more important that event the path to the bam files are different, the main bam file name are same → it is not accepted → need to change the name of the bam files to make it unique (<https://www.biostars.org/p/119795/>)

-It is very easy to mis-input the chromosome size (hg19 or mm10) → can not check the results to tell if we are wrong → be careful

-Input sample is not required for SF calculation

-Vẫn chưa hiểu vì sao đoạn cuối của SIF báo lỗi ko có results

## Datasets the authors used in SpikeInFree

-Check lại paper SpikeInFree: trong 67 dataset trải dài 5 loại histone modifications, thì có:

+30 là H3K27me3

+11 là H3K27Ac

+9 là H3K4me3

+4 là H3K79me2

+12 là H3K36me3

## Density plot (used in SpikeinFree paper):

<https://datavizcatalogue.com/methods/density_plot.html>

-Giống như histogram nhưng independent với number\_of\_bin

-Cung cấp cái nhìn rõ hơn về distribution shape

## How to run an R-script on Computerome

Rscript scriptFilename.R

Nếu load R theo version nào thì sẽ được dùng version đó, nếu có conflict giữa các module, thì thoát ra khoỉ computerome, log vô lại là giải quyết được.

## Some Linux commands that are useful

-echo -n: not enter a new line

-Ctrl-L: clear screan

## Working with gtf file

-Need to ensure that the chromosome names in the gtf file are corresponding to those from fastq files

-Sites to download: i.e. for fly genome: <http://ftp.ensemble.org/pub/release-104/gtf>

-Gtf file should be downloaded from gencode database because it is updated monthly

## htseq-count

-Pay attention to -t and -i parameters

+Should use -t with gene

+Should use -i with gene\_id (better than transcript\_id and gene\_name)

-If the input bam file of ht-seq count is not indexed, there will be a warnings but the command can still run and generate the same results

-Should not use gtf file from Genome browser as the gene\_id and transcript\_id seems the same (from FAQS of htseq-count).

## Broad Institute manual

Contain explaination for some basic commands such as deeptools, htseq, picard

+ Khi dùng bamCoverage hoặc bamCompare, mà data cần phải filter thì cần áp dụng các lệnh filter trước vì các lệnh bamCoverage hoặc bamCompare của deeptools có bước filter nhưng được thực hiện sau khi scaling factors được thực hiện

## Samtools sort command creates a lot of tmp files

Reason: https://github.com/samtools/samtools/issues/603

## Resource to learn sed/awk quickly

- http://reasoniamhere.com/2013/09/16/awk-gtf-how-to-analyze-a-transcriptome-like-a-pro-part-1/

- From minute 1:15:00 of this video (in Vietnamese)

https://www.youtube.com/watch?v=JPwVjOVLNKc&list=PLXtgXP89Tyn-aRmrnfNsio7LvG1BWy9eb&index=11

-Notes:

-Điều đầu tiên cần nhớ là nó phù hợp cho ai dùng nhiều với text data

-awk linh động và giải quyết vấn đề hệ thống hơn sed

-Nhiều khi, dùng sed là đủ vì vấn đề ko quá phức tạp đến mức cần dùng awk

-sed:

+stream editor

+có thể dùng để edit nhiều thứ trong nhiều file trong 1 câu lệnh --> powerful

+là lựa chọn tốt khi ko thể mở large file trong interactive để edit

## Change cache folder for using sra toolkits

By default, when running sra toolkits, the temporaty file will be created in cache folder in your home directory in Computerome which is limitted to 10GB --> Use the guidance from this link to change to your desired folder: https://standage.github.io/that-darn-cache-configuring-the-sra-toolkit.html

## Proper way to download paired-end raw sequencing data (fastq files) from NCBI

Run this script on computerome

module load sratoolkit/2.8.1

module load edirect/7.50

prefetch -v SRR5331796 -O /home/projects/cu\_10056/data/Yang/MitoNormalization/sra\_fastq/Paired\_end

fastq-dump -A SRR5331796 --split-3 -O /home/projects/cu\_10056/data/Yang/MitoNormalization/sra\_fastq/Paired\_end/sra\_fastq

## Notes that include a lot of concepts

20211115-Note --> cần check lại

20211209-Note

20211210-Note

20211214-Note

20211217-Note

20211220-Note

20211228-Note

20211229-Note

20211230-Note

20220104-Note

## Rules to process TT-seq data

TT-seq data is one kind of RNA-seq data

-If it is single-end sequenced: don’t remove duplicated reads

-If it is paired-end sequenced: remove duplicated reads

Considerations:

-Does the TT-seq data care alternative splicing?

-Is it single- or paired-end?

-Read is stranded or not?

-Need to understand reverse and forward strand

-Need to understand every parameters the authors use

## Sam files

-Mapping using bwa mem command:

-Single-end reads: 1 input file is needed

-Paired end reads: 2 input files are needed

→ in both case, bwa mem generate only one output sam file

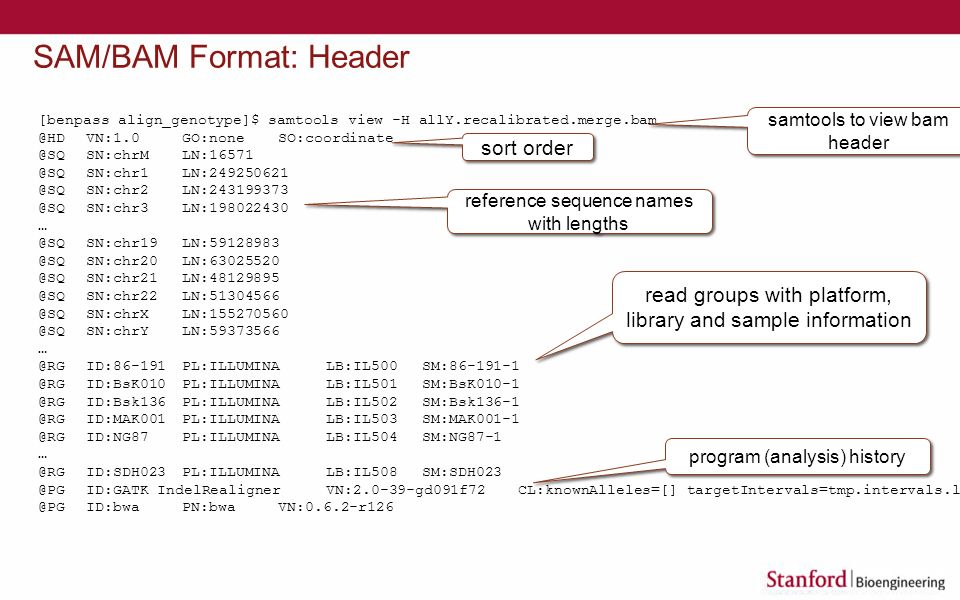
-Using command: cat samfile | cut -f1 | sort | uniq | wc -l → compare with the number of reads in the raw fastq file → see how many reads are mapped and unmapped

-Check the number of unmapped reads: samtools view -f 4 samfile | wc -l

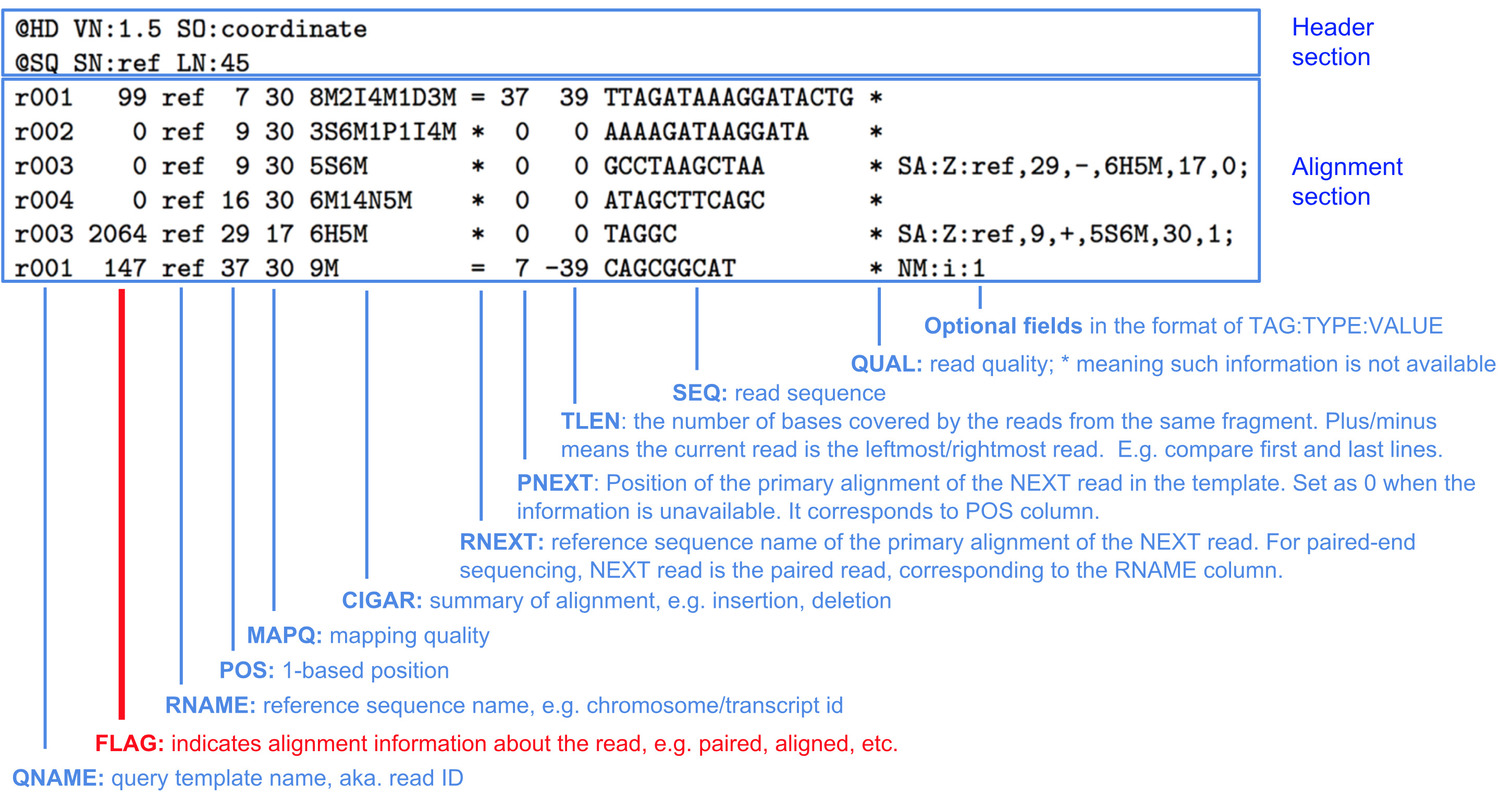
-Check the number of mapped reads: samtools view -F 4 samfile | wc -l. The number of mapped readed might greater than the number of original reads → one read can be mapped 0 to n times (n>=1)

-Website to check sam format flag: https://www.samformat.info/sam-format-flag

-Sam/Bam header format



-Sam/Bam alignment section format



## Duplicated reads

Are reads that come from the same DNA fragments

## Understanding FastQC report from manual

Tìm hiểu FastQC từ <https://dnacore.missouri.edu/PDF/FastQC_Manual.pdf>

- Mục tiêu là để đảm bảo dữ liệu:

+looks good

+ko có problems

+Ko có bias

- Tool FastQC:

+tốt hơn report của sequencer vì nó report problem xuất hiện từ sequencer lẫn problem do library material

+Input có thể là FastQ, SAM, BAM

+Theo hỏi đáp ở đây thì fastQC có support paired-end reads nhưng sẽ analyse forward and reverse reads separately

- Đánh giá kết quả từ FastQC:

+Cần căn cứ vào ngữ cảnh, mình muốn kết quả ntn từ library của mình

+Kết quả được xem là normal nếu nó random và diverse. Nếu có bias trong kết quả thì có thể nó là particular way we expect from our library

+Nên xem summary report là 1 pointer mà tại đó mình cần tập trung và hiểu vì sao library của mình ko random và diverse

+Chú ý có 2 loại report là per base và per sequence

- Những nội dung đáng chú ý từ FastQC report:

+Basic statistics:

-Ko bao giờ có Warning hoặc Error

-Từ đây có thể biết được độ dài của reads

-Từ đây có thể biết được có tổng bao nhiêu sequence/read, bao nhiêu được flag và filter

-Từ đây biết được input vào sequencer là raw base (actual base calls) hay là colorspace data mà sẽ từ đó được convert thành base call

+Per base sequence quality

-Càng về cuối thì quality giảm dần

-Màu green là tốt

-Hai đầu của whiskers thể hiện 10% và 90% points

-Blue line thể hiện mean quality

-Yellow box thể hiện inter-quartile range (25-75%)

-Warning: nếu có bất kì base nào có lower quartile thấp hơn 10 hoặc median của bất kì base nào thấp hơn 25

-Failure: nếu hai chỉ số trên là 5 và 20

+Per sequence quality scores

-Biểu đồ này để ta biết được có subset sequence nào đó có base call quality chất lượng thấp

-Warning: nếu mean quality per read thấp hơn 27, tương đương 2% error rate

-Failure: nếu chỉ số trên là 20, tương đương 1% error rate

+Per base sequence content

-Biểu đồ biểu diễn content (tỉ lệ ATGC) trên từng base một. Tốt nhất là các đường này song song với nhau, thể hiện ko có bias cho bất cứ loại base nào

-Warning: nếu tỉ lệ chênh lệch của A với T hoặc G với C là 10% cho bất cứ vị trí nào

-Error: nếu tỉ lệ trên là 20%

+Per base GC content

-Tốt nhất nên là 1 đường ngang song song Ox

-Nếu đường này ko nằm ngang mà thay đổi --> trong data có overrepresented sequence --> contaminate the library

-Warning: nếu có vị trí nào lệch 5% so với mean GC content

-Error: nếu có vị trí nào lệch 10% so với mean GC content

-Trong ví dụ của mình ko thấy có report này

+Per sequence GC content

-Được biểu diễn dưới dạng distribution và so sánh với standard distribution với mean là 50%.

-Nếu hình dạng có sự khác biệt: ngụ ý contaminated library hoặc một loại biased subset nào đó

-Nếu distribution bị ship: systematic bias mà ko phụ thuộc vào vị trí của base

-Warning: nếu tổng deviation chiếm 15% of the read

-Error: nếu tổng deviation chiếm 30% of the read

+Per base N content

-Nếu sequencer ko chắc về 1 base nào đó --> dùng kí tự N

-Thông thường thì sẽ có low proportion là Ns, đặc biệt là lúc gần cuối sequence

-Warning: 5% ở bất cứ vị trí nào

-Failure: 20%

+Sequence length distribution

-Warning: nếu sequence length ko bằng nhau

-Failure: nếu có sequence length bằng 0

+Duplicate sequences

-Diverse library (good library) có nghĩa là mỗi sequence gần như là duy nhất.

-Low level of duplication: dấu hiệu của high level of coverage of the target sequence

-High level of duplication: dấu hiệu của enrichment bias (ví dụ PCR over amplification)

-Với RNA sequencingsequencing, you are sequencing a hugely enriched pool of stuff. With mRNA sequencing, it represents only 1 to 2% of the genome. There are certain signals, there are certain mRNAs in that pool of data, that are hughly enriched. There are certain genes that are expressed at very high levels that will produce a reasonable amount of the data that we’re seeing. So you expect to see duplication in RNA-seq experiments. So there is only about 30% of the reads overall are unique which is high level of duplication.

+Overrepresented sequences

-Can relate this section to Duplicate sequence. If there is high level of duplicate sequence → there are high level of overrepresented sequences

-As long as none of the Possible source for hit is adaptor, it probably ok for RNA-seq experiments

+Adapter content

## Notes about genomic interval

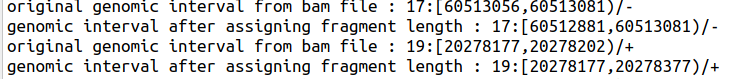
+ Trong hai giá trị start và end thì start luôn là giá trị nhỏ hơn, để cả khi đó là ở reversed strand

+ Nếu length được gán thì:

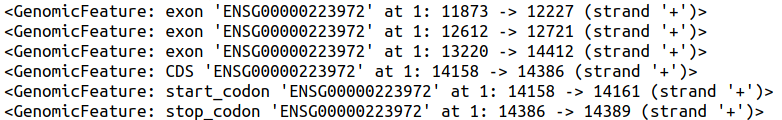
-Với + strand thì start giữ nguyên, end thay đổi, end=start+legnth

-Với - strand thì end giữ nguyên, start thay đổi, start=end-length

Ví dụ với length=200



## Example of features (from GFF parser):



## Some notes about sed/awk commands

- find -name "\*.matrix" | xargs wc -l: tìm các tập có đuôi là matrix, xong đếm số dòng ở mỗi tệp

- cat gencode.Grch37.basic.annotation.gtf | grep -B1 -A2 ENSE00003582793.1\_1: tìm pattern và trả về luôn cả before và after của nó

-Lệnh sed dùng để in ra dòng cụ thể

+sed -n '3,5'p sed\_example.csv: in ra dòng thứ 3 và 5 của file

+sed -n '5'p sed\_example.csv: chỉ in dòng thứ 5

-Lệnh sed dùng thay cho grep

+sed -n '/John/'p sed\_example.csv

+sed -n '/John/,/Jason/'p sed\_example.csv

+grep "John\|Jason" sed\_example.csv

-Lệnh sed dùng để thay thế, có hai kiểu là print output to sreen, hoặc change the file in-place (using option -i)

+sed 's/Developer/Coder/' sed\_example.csv

+sed -i 's/Developer/Coder/' sed\_example.csv --> thay đổi nội dung file sed\_example.csv

++sed -ibackup 's/Developer/Coder/' sed\_example.csv --> thay đổi nội dung nhưng cũng tạo ra file backup

-Lệnh sed để xóa kí tự trắng và dòng trống

+'s/\s//g' sed\_example.csv: sử dụng regular expression: s/: substitute, \s: khoảng trắng, // kí tự rỗng, g: global

+'s/ //g' sed\_example.csv: công dụng như trên nhưng không dùng Regular expression

+sed '/^$/d' sed\_example.csv: xóa dòng trống, ^:đầu dòng, $:cuối dòng, d: xóa

+awk: là một ngôn ngữ lập trình, có biến,.. làm được nhiều việc hơn lệnh cut. Awk là cut với điều kiện. Hơn nữa, awk có thể process file và có thể dùng kết hợp với các lệnh khác như loop, set variable

+cut -f 1,2,3 awk\_example.csv

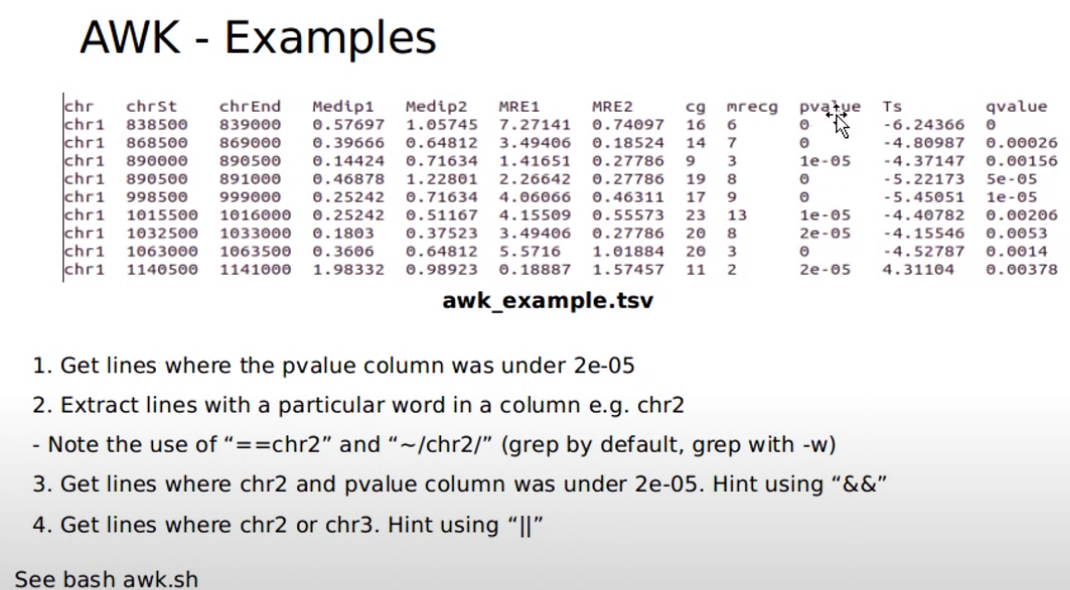
+awk -F"\t" '{OFS="\t"} {print $1, $2, $3}' awk\_example.csv:

-F"\t": filed được phân cách bởi \t

{OFS="\t"}: output cũng được phân cách bởi \t

print $1, $2, $3: in ra 3 biến, phân cách theo \t

-Các bài tập dùng awk:

- Lệnh để biết các cột là cột thứ mấy --> pvalue là cột thứ 10

head -1 awk\_example.csv | tr "\t" "\n" | nl

- Đếm số dòng của file: wc -l awk\_example.csv

-In ra những dòng có pvalue>0:

Chú ý: $0 là tất cả các cột

+awk -F"\t" '$10>0 {print}' awk\_example.csv | column -t: in ra tất cả các cột

+awk -F"\t" '$10>0 {print $1, $2, $3 }' awk\_example.csv | column -t: như trên nhưng chỉ in ra 3 cột đầu

-In ra những dòng có chính xác là chromosome thứ hai

+awk -F"\t" '$1=="chr2"' awk\_example.csv

+awk -F"\t" '$1=="chr2",1' awk\_example.csv: chưa hiểu ,1 kia giống như $0

+awk -F"\t" '$1=="chr2"' awk\_example.csv | sort | uniq -c: kiểm tra lại kết quả trên

+awk -F"\t" '$1=="chr2" {print $1} ' awk\_example.csv | sort | uniq -c: tương tự cũng để kiểm tra lại kq

-In ra những dòng có chr2 (gồm cả chr2, chr21, chr22)

+awk -F"\t" '$1~/chr2/ ' awk\_example.csv | column -t

-Tương tự trên grep

+grep chr2 awk\_example.csv: in ra những dòng có chr2 (gồm cả chr2, chr21, chr22)

+grep -w chr2 awk\_example.csv: in ra những dòng có chính xác chr2 thôi

-Kết hợp các điều kiện: || là or, && là and

+awk -F"\t" '$1=="chr2"||$1=="chr21" {print $0} ' awk\_example.csv

-Dùng awk để vừa cut vừa process file (trong lệnh này có sử dụng built-in variable của awk là NR: num\_row)

+awk -F"\t" '{OFS="\t"} NR>1 {print $1, $2, $3, $3-$2}' awk\_example.csv | column -t

+(echo -e "chr\tchrSt\tchrEnd\twidth" ; awk -F"\t" '{OFS="\t"} NR>1 {print $1, $2, $3, $3-$2}' awk\_example.csv )| column -t: cũng lệnh trên nhưng bổ sung thêm header

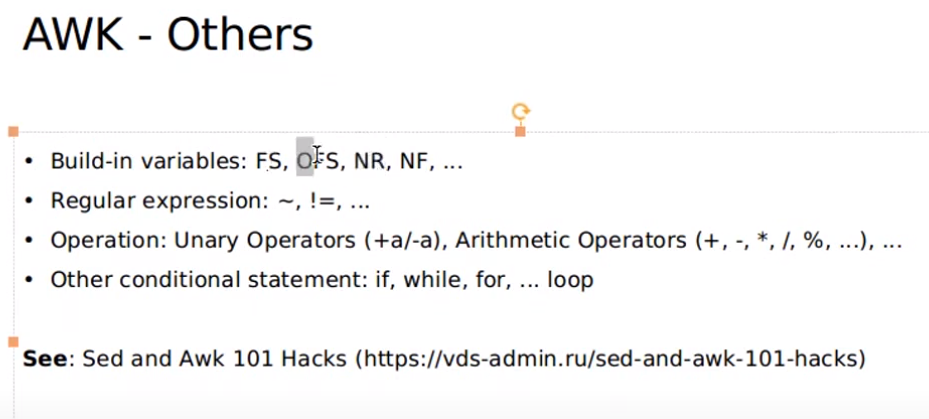
-Dùng awk để process phức tạp hơn:

+awk -F "\t" '{OFS="\t"} NR>1 {if ($8/($3-$2)>0.05) print $1,$2,$3,$3-$2,$8/($3-$2),"High"; else print $1,$2,$3,$3-$2,$8/($3-$2),"Low" } ' awk\_example.csv | column -t

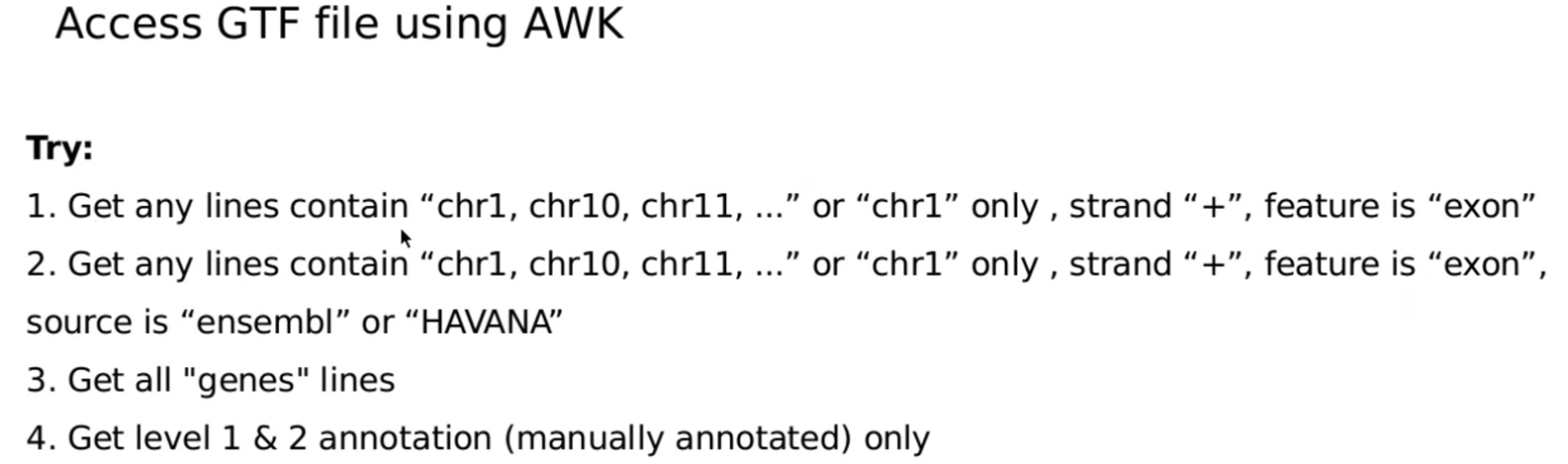
-Dùng awk để thay thế có điều kiện (tốt hơn sed) với gsub:

+awk -F"\t" '$1==105 {gsub ("Sales Manager","Employee", $0); print } ' sed\_example.csv: thay thế nhân viên có ID =105

-Thêm về awk:



-awk và gtf file

Câu 1: awk -F"\t" '$1~/chr1/ && $7=="+" && $3=="exon" {print $0}' sample.GTF

hoặc awk -F"\t" '$1~/chr1/ && $7~/\+/ && $3=="exon" {print $0}' sample.GTF: thêm \ trước dấu + để nó ko trở thành regular expression

Câu 2: awk -F"\t" '$1~/chr1/ && $7=="+" && $3=="exon" && ($2=="HAVANA" || $2=="lincRNA") {print $0}' sample.GTF

Câu 3: awk -F"\t" '$3=="gene" {print $0}' sample.GTF

More on sed command: <https://vds-admin.ru/sed-and-awk-101-hacks>

Câu 4: Tính chiều dài mỗi chromosome trong fa file

awk -v RS=">" 'NR>1 {print $1 ":" length($0)}' mm10\_wM.fa

RS: record separator, NR: num row

- Lệnh if trong shell:

Link hữu ích: <https://ryanstutorials.net/bash-scripting-tutorial/bash-if-statements.php>

+Ví dụ kết hợp lệnh if và grep, ls, echo

for i in `ls \*.GTF`; do if grep -q "exon" $i ; then echo $i ; fi; done

-Oneliner command: <https://linuxcommandlibrary.com/basic/oneliners.html>

-Link để luyện tập: [https://bioinformaticsworkbook.org/#gsc.tab=0](https://bioinformaticsworkbook.org/" \l "gsc.tab=0)

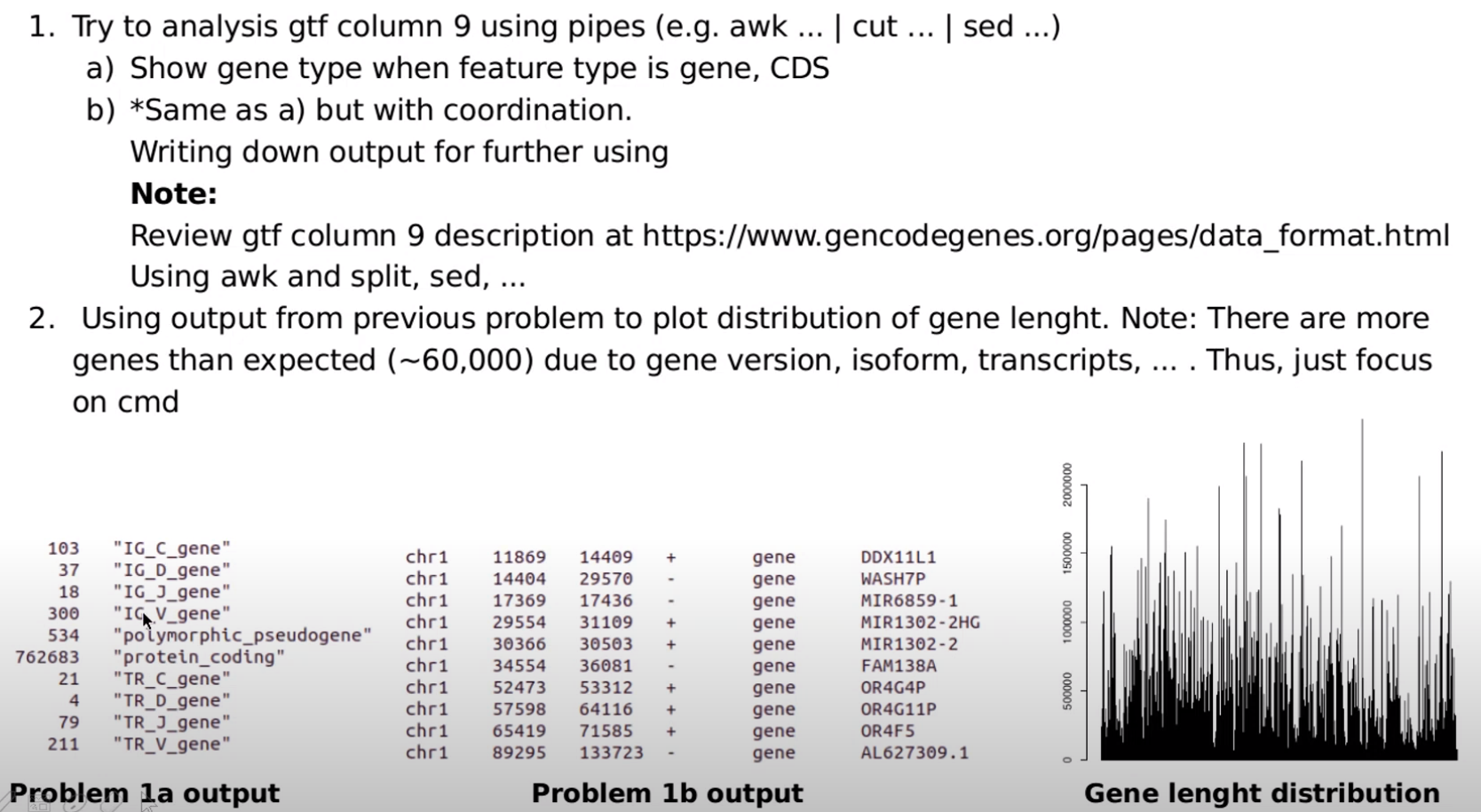
-Combine bash script với R

cat gencode.Grch37.basic.annotation.gtf | grep -v "#" | cut -f3 | sort | uniq -c | Rscript -e '

dt <- read.table("stdin", header=F);

par(mar=c(7,7,7,7));

barplot(dt$V1, names.arg=dt$V2, las=1, horiz=T)'

-Lệnh lấy thông tin gene type khi feature là gene, CDS

cat gencode.Grch37.basic.annotation.gtf | awk -F"\t" '$3~/CDS/{print $9}' | cut -d";" -f3 | sort |uniq -c | sed 's/gene\_type//'

-Lệnh lấy thông tin gene name

cat gencode.Grch37.basic.annotation.gtf | awk -F"\t" '{OFS="\t"} $3=="gene" {split($9,a,";"); print $1,$4, $5, $7, $3, a[3] }' | sed -e 's/gene\_name//g' -e 's/\"//g' > gene\_name.tsv

## RNA-seq revision

1. Vì sao cần làm RNA-seq:

Để tìm biểu hiện của gene giữa các điều kiện khác nhau

2. Platform để thực hiện analysis tương tự ngoài RNA-seq

RNA expression array

Tuy nhiên RNA-seq cung cấp nhiều thông tin hơn, tìm được biểu hiện của de novo gene

3. Challenge trong RNA-seq

RNA dễ bị phân mảnh, ảnh hưởng bởi mồ hôi tay của người làm thí nghiệm

mRNA chỉ chiếm 5%

4. Khi làm RNA-seq thì ta thường cần bao nhiêu replicate

Genome nhỏ: 20 rep, genome lớn như người: 2-3

Vì sao cần rep: để hiểu technical bias, batch effect, con-founder

5. Whole genome sequencing cần bao nhiêu X để call SNP?

RNA-seq thông thường cần 10X, trong khi với variant calling cần 30 X → RNA-seq để làm Differential expression analysis cần >30 X

6. mRA trong nhân khác với mRNA ngoài nhân ntn?

mRNA trong nhân: vẫn còn intron và chưa gắn poly A và trải qua quá trình splicing mà quá trình này là cell-sepcific (còn gọi là Pre-mRNA)

mRNA: sẵn sàng cho sequencing và translation

Chú ý: ngoại trừ repeat, thì chỉ cần 25 nu trở lên thì seq đã rất specific rồi

7. Các gene liên quan đến ti thể rất nhiều, đặc biệt với các tế bào hoạt động mạnh nhw gan (gan thường có khoảng 2000 ti thể)

## RNA-seq workflow and applications

-Using RNA is that we isolate the RNA, especially people are focusing on mRNA in our tissue and we were trying to take a snapshot of the tissue

-Current limitation does not allow to profile all the proteins → we instead profile the RNA

-Why don’t we do microarray? Because RNA-seq allows you to discover novel transcripts and broaded range of dynamic detection (same genes can be detected in this sample but not in other samples). RNA-seq also has higher detection sensitivity and specificity

-Experimental design

+A disease in dogs caused by a mutation

+However, some dogs have slow progression while other have fast progression although they have the same mutation

+We can use RNA sequencing (RNA-seq) to study:

--Differentially expressed genes

--What are the Pathways that these gene involved?

+For analysis part: 3 questions to be asked

--Which genes do these reads belong to?

--How many reads align to a specific gene?

--Do different sample groups express genes diffentially?

+Tools for workflow, should go together in a suite

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tuxedo suite | | | | |
| 1-Quality control | Fasta, GTF | Fasta, GTF | Fasta, GTF |  |
| 2-Alignment | Tophat (.sam, .bam) | HISAT2 | HISAT2 | Which genes do these reads belong to? |
| 3-Transcript assembly | Cufflinks | StringTie | HTSeq (.txt) | How many reads align to a specific gene? |
| 4-Differential expression | Cuffdiff | Ballgown | DESeq2 (R/Rstudio) | Do different sample groups express genes diffentially? |

\*Yellow highlight: in supercomputer

+Reference genome: ranked

--www.ensembl.org: don’t need to transfer its ID into another system

-Type in the search keyword, i.e. dog genome –< ftp link

--UCSC

+DeSeq2 or EdgeR

--DeSeq2 > 12 replicate

--EdgeR: < 12

However, they don’t differ a lot

+Downstream analysis

--PCA: to group genes according to time-resolved experiment or conditions (fast, slow, control)

--Heatmap

--Venn diagram

+Pathway analysis

--Panther

--IPA

**RNA-seq applications:**

The main goals of RNA-seq are to identify:

-The sequence: particular order of A, C, G, U residues

+Can be used to identify known protein coding genes, novel genes, long non-coding RNAs.

+Gene model: a collection of exons and introns that make up a gene

+RNA-seq: can maps to

--known protein-coding genes

--Completely novel exons

--5’TSS, 5’UTR, exon-intron boundaries...

-The structure: location of promoter, intron-exon junctions, 5’ and 3’ untranslated regions, polyA sites. Secondary structures, tertiary structure: 3D shape of the RNA molecular

-Abundance: numerical amounts of reads of each particular sequence both as absolute and normalized values.

## Long non-coding RNA (lncRNA)

- Definition: are transcript of length > 200 nu that are characterized by a low coding potential

- The choice of length threshold is arbitrary but serves as a way to separate it with other non coding RNA such as microRNA, small nucleolar RNAs, …

- Total RNA – mRNA → non coding RNA → filter by length → assessing coding potential→ long nc RNA

-Long nc RNA lack strong interspecices conservation

-Người ta phân loại long nc RNA theo vị trí như: intronic, intergenic, divergent, anti-sense

-Trên lncRNA vẫn có splicing nhưng chỉ là inefficient splicing và ko có quá trình translation như đối với mRNA

-Những chức năng quan trọng của lncRNA:

+regulate chromatin

+transcription regulation

## Other useful sources to learn bioinformatics

[https://bioinformaticsworkbook.org/list.html#gsc.tab=0](https://bioinformaticsworkbook.org/list.html" \l "gsc.tab=0)

https://userweb.eng.gla.ac.uk/umer.ijaz/bioinformatics/linux.html

## Flag information of reads

From: http://seqanswers.com/forums/showthread.php?t=15280

-Single end reads yield only 3 possible flag values: 0,4 and 16.  
 -0 means the read aligned in the forward direction. 16 mean it aligned in the reverse direction. 4 means it didn't align.  
 -The rest of the flags are for paired end data only.

-Flag 0 means "the read is not paired and mapped, forward strand"

The SAM flag field although it appears as a single number actually contains several pieces of information which have been combined together. It is a bitwise field, which means that it makes use of the way that computers represent numbers to store several small values stored in one large value.  
  
If you think of a standard integer as being composed of 32 bits (0 or 1) then it would look like:  
  
00000000000000000000000000000000  
  
However SAM uses this single number as a series of boolean (true false) flags where each position in the array of bits represents a different sequence attribute  
  
Bit 0 = The read was part of a pair during sequencing  
Bit 1 = The read is mapped in a pair  
Bit 2 = The query sequence is unmapped  
Bit 3 = The mate is unmapped  
Bit 4 = Strand of query (0=forward 1=reverse)

**Hôm nay mới nhận ra là miễn là flag là số lẻ thì nó là paired-end!**

## Important note about installing with global and local

|  |  |
| --- | --- |
| Global | Local |
| $ sudo pip install:  installs the package globally in your python installation, i.e. for all users | $ pip install –user:  installs to the local user directory, i.e. ~/.local/lib.python – just you |
| Example:  $ sudo pip install jupyter  $ jupyter notebook  → Will run jupyter, open a web browser, allow you to work with notebooks  Notes: there were recently malicious code included in pypi. Never use sudo to install with pip. This is the same as running a virus as root. Either add your local folder to your PATH or use a virtualenv | Example:  # pip install –user jupyter  # jupyter notebooks  → Will do nothing until your local directory has been added to your PATH |
| /usr/bin  -Belongs to the system. Mess with it at your peril. If you trash /usr/bin, you'll probably end up reinstalling the OS.  -Is a system folder containing binary files | /usr/local/bin  --Is yours to fool with; if you mess something up in there, you can trash /usr/local and the system will chug along just fine.  -Is a local folder. |

## Notes about using virtual environment

**Link TUT:** **https://www.biostars.org/p/289547/**

- Loay hoay mãi, ko hiểu sao cài mà ko chạy được lệnh mkvirtualenv, thì là do cần phải cài virtualenvwrapper, cài xong cần phải tìm nó bằng lệnh

find / -name virtualenvwrapper xong đưa đường dẫn vào ~/.bashrc xong source ~/.bashrc thì mới được (Nguyên nhân chính là do trên các version khác nhau của Ubuntu, nó có thể được cài vào /usr/local/bin/virtualenvwrapper.sh hoặc ~/.local/bin/virtualenvwrapper.sh) nên cần tìm đường dẫn chính xác để source).

sudo pip install virtualenv

sudo pip install virtualenvwrapper

sudo find / -name virtualenvwrapper.sh

gedit ~/.bashrc →

export WORKON\_HOME=~/virtualenvs

export VIRTUALENVWRAPPER\_PYTHON=/usr/bin/python3

source /usr/local/bin/virtualenvwrapper.sh

source ~/.bashrc: reload profile mới vừa được edit

mkvirtualenv -p /usr/bin/python3 AP\_prepare: env được tạo trong thư mục cài đặt /home/trung/virtualenvs, tạo xong, activate env luôn

virtualenv env: env được tạo trong thư mục hiện tại

-Chú ý thứ hai là find / -name: tìm tên file từ root hiệu quả hơn which

-Thoát khỏi virtualenv thì dùng lệnh: deactivate dù là loại env nào, muốn activate lại thì

+ Với env được tạo bằng mkvirtualenv: dùng lệnh workon tên\_env

+ Với env được tạo bằng virtualenv: dùng lệnh source tên\_env/bin/activate

-Tạo danh sách các package và dependencies phục vụ mục đích replication:

pip freeze --local > requirements.txt

-Ko dùng env nữa thì xóa thư mục đó đi bằng rm -rf

-Một số lưu ý:

+activate chỉ có hiệu lực bên trong 1 terminal session, nếu đóng terminal mà quên ko deactivate thì cũng chả sao cả

+Nếu muốn thay đổi env, chỉ cần activate env mới mà ko cần deactivate cái cũ

+Virtual environment và environment variables là hai thứ khác nhau

**+Nguyên nhân cần phải có các virtual environment khác nhau cho các project khác nhau là để các packages và dependencies của các project này ko bị lẫn với nhau, tránh các environment variables bị override với nhau**

## Review about Anaconda, Conda

Anaconda là:

+Một open source Python distribution mà giúp việc up and running (operation) các packages và libraries của Python dễ dàng hơn

Anaconda navigator:

+Là GUI được tự động cài cùng Anaconda

Conda:

+Là package manager của Anaconda

+Tương tự như pip của Python nhưng nhiều khi, cài đặt thông qua Conda thì dễ hơn với pip

+Ko nhất thiết là replacement của pip, ta có thể dùng pip 99% và cài những package không cài được với pip bằng conda

+Có thể quản lý được virtual enviroment

Quy trình tạo và quản lý env hiệu quả với conda

1.Tạo folder chứa toàn bộ dự án

2.Chuyển đến folder đó

3.Tạo env bằng lệnh: conda create --name env\_name tên các dependencies (ví dụ flask, sql..)

4.Activate env bằng lệnh: source activate env\_name. Chú ý, khi tạo biến mt xong thì có thể cần phải init nó trước bằng lệnh conda init bash (tên shell), và khởi động lại terminal.

Nếu muốn deactivate: source deactivate env\_name

5.Trong môi trường conda, vẫn có thể dùng lệnh pip list để xem danh sách các packages ta đã install.

6. Khi dự án kết thúc, để giúp dễ dàng replicate, ta có thể tạo file lưu danh sách các package cùng với version, và lưu vào file .yaml bằng lệnh:

conda env export > environment.yaml

7. Khi muốn install lại các package trong environment, ta dùng lệnh

**conda env create -f environment.yaml**

## Managing environment variables

-Các biến môi trường thường được dùng trong nhiều dự án để lưu database URILs, lưu secret key, Python path…Nó cho phép ta access sensitive data mà ko cần đặt nó trực tiếp vào project

-Global env variables được share trong nhiều project nhưng ta cần đảm bảo chúng ko overwrite lẫn nhau

-Liệt kê các env: conda env list

-Chọn 1 env, chuyển đến thư mục chứa env đó

-Tạo 2 directory, 1 trong đó sẽ chứa script mà sẽ chạy bất cứ khi nào env này được activated/deactivate là etc/conda/activate.d/env\_vars.sh và etc/conda/deactivate.d/env\_vars.sh